unit; i.e., a memory unit in which the bit positions for a program command are output successively bit by bit. In one refinement, the storage capacity of the bootstrap memory unit is chosen to be much lower than the storage capacity of the reload memory unit. Thus, the storage capacity of the bootstrap memory unit is in the kilobyte range. As bootstrap memory unit, serial-access EEPROMs (Electrical Erasable Programmable Read Only Memory) are suitable; such circuits operate with an IIC bus system, for example.

In yet another embodiment, the main memory unit is a volatile memory unit. The main memory unit permits simultaneous input or output of a number of bit positions of a program command. As such, a processor can access the main memory unit directly. The main memory unit used can be a RAM (Random Access Memory). Synchronously operating dynamic RAMs permit short access times.

In a further embodiment, the reload memory unit is a nonvolatile memory unit. Reload memory units which output the stored program commands serially can be used. In one development, the storage capacity of the reload memory unit is in the megabyte range. The reload memory unit can, therefore, store operating systems having several hundred megabytes. In one refinement, the reload memory unit used is a "multimedia card". To access such cards, a protocol needs to be observed which the processor does not control and which uses specific control commands. The control commands include data words whose bits stipulate the command to be executed. The commands have been standardized by the MMC agreement (Multi Media Card). More detailed information relating to such cards can be obtained at www.scandisc.com.

A Compact Flash card and/or a SmartMedia card and/or a Memory Stick memory unit (e.g., from Sony) are also used, however. Some of these storage media output, by way of example, four bits in parallel, so that a number of read operations are required in order to access a program command having more than the bit positions respectively output in parallel.

In a subsequent embodiment, the reload memory unit contains a register in which the start address of a currently readable memory area of the reload memory

unit is noted. Another memory area, whose start address is currently not noted in the register, cannot be read. When the bootstrap program is executed, the reload transfer operation is executed on the basis of the start address; i.e., only the data words stored in the readable memory area are accessed. The use of, by way of example, multimedia cards having such registers allows another memory area to be stipulated easily; namely, by entry of another start address in the register.

In one embodiment, the simple selection option for memory areas in the reload memory unit is utilized in order to update program commands. A new version of the program commands can be stored in the currently unreadable memory area. If the data processing installation needs to be restarted during the storage operation, the restart operation can be performed irrespective of the state of the memory for the new version. This is because the old version still exists in full in the reload memory unit. Only after the new version of the program commands has been fully transferred to the reload memory unit is the address which is noted in the register set to the other memory area. A new start operation is then initiated. If errors arise during this start operation, the old start address is entered in the register again. Starting is then repeated. In the memory unit, the previously used, tried-and-tested program commands are then used for the new start operation. This measure can prevent the data processing installation from being unavailable for a relatively long time when changing to a new version of the program commands. This is particularly necessary for telecommunications installations or for exchanges, because high demands are placed on the failure reliability thereof.

In another embodiment, at least once during execution of the program commands transferred to the main memory unit, at least one portion of these program commands is copied from an original area in the main memory unit to another area of the main memory unit. A jump command can be used to make the processor start executing the program commands stored in the other memory area. Commands to be transferred from the reload memory unit are then stored in the original area, where they overwrite the previously stored program commands. After the current transfer operation, the processor is then switched to a defined initial

state; e.g., using the control unit. This measure ensures that, after the transfer process is complete or after a portion of the transfer process is complete, execution of the program commands transferred during the transfer process can be started easily. This is because such a defined initial state has been prescribed for commercially available processors. By way of example, this initial state is adopted when a reset signal is produced. In the initial state, the registers of the processor have prescribed values, and command execution starts at an address prescribed by the manufacturer of the processor. Complex measures for prescribing register contents and processing areas can thus be avoided.

In one embodiment of the inventive process, the reload memory unit stores program commands using a compression process. The use of a compression process allows the memory space required in the reload memory unit to be considerably reduced. By way of example, only less than one third of the memory space is now required. Program commands for carrying out the associated decompression process can be stored in the bootstrap memory unit or in an uncompressed portion of the program commands in the reload memory unit. When the program commands of the decompression process are executed, the compressed program commands are decompressed.

In one embodiment, after the bootstrap transfer operation, the bootstrap program is copied over within the main memory unit. The bootstrap program is then executed, the original area being overwritten by program commands from the reload memory unit. The processor is then put into a defined state for the second time. The program commands transferred previously from the reload memory area then start to be executed; e.g., execution of the operating system is started.

In another embodiment, the processor is put into a defined state three times during the restart operation. Firstly after the bootstrap transfer operation, after an auxiliary load transfer operation in which the decompression program is transferred, and then at the end of the reload transfer process for starting the operating system. This practice makes it a simple matter to switch between the individual stages.